Amendments to the Claims

This following listing of claims will replace all prior versions and listings of claims in the application.

Claim 1 (original) A method of generating an intermediate representation of program code written for running on a programmable machine, said method comprising:

(i) generating a plurality of register objects for holding variable values to be generated by the program code; and

(ii) generating a plurality of expression objects representing fixed values and/or relationships between said fixed values and said variable values according to said program code;

wherein at least one variable sized register is represented by plural register objects, one register object being provided for each possible size of the variable sized register.

Claim 2 (original) A method according to claim 1, wherein a write operation to a variably sized register is effected by writing to the register object corresponding to the appropriate size and maintaining a record of which register objects contain valid data.

Claim 3 (original) A method according to claim 2, wherein a read operation from a variably sized register is effected by determining from said record if there is valid data in more than one corresponding register object which must be combined to give the same effect as reading from the variably sized register, and

(i) if it is determined that no such combination is required, reading from the appropriate register object; and

(ii) if it is determined that such combination is required, combining the contents of appropriate register objects to provide a read value.

Claim 4 (original) The method of claim 1, comprising translating the program code written for execution by a processor of a first type so that the program code may be executed by a processor of a second type, using the generated intermediate representation.

Claim 5 (original) The method of claim 4, wherein the translation is performed dynamically as the program code is run.

Claim 6 (original) The method of claim 1, comprising optimising the program code by optimising said generated intermediate representation.

Claim 7 (amended) The method of claim 6, wherein the optimizing step is used to optimize the program code written for execution by a processor of a first ~~pipe~~ type so that the program code may be executed more efficiently by that processor.

Claim 8 (original) A method of generating an intermediate representation of program code expressed in terms of the instruction set of a subject processor comprising at least one variable sized register, the method comprising the computer implemented steps of:

generating a set of associated abstract register objects representing the variable sized register;

for each write operation of a certain field width to the variable sized register, writing to an abstract register of the same width;

maintaining a record of which abstract register objects contain valid data, which record is updated upon each write operation; and

for each read operation of a given field width, determining from said record whether there is valid data in more than one of said different sized abstract registers of the set which must be combined to give the same effect as the same read operation performed upon the variable size register; and

(a) if it is determined that no combination is so required, reading directly from the appropriate register; or

(b) if it is determined that data from more than one register must be so combined, combining the contents of those registers.

Claim 9 (original) The method according to claim 4, wherein the step of determining whether or not the contents of more than one abstract register must be combined and if so which abstract registers must be combined, is determined in accordance with the following conditions in respect of each set of different sized abstract registers:

(i) if the data required for an access lies wholly within one valid abstract register, that register only is accessed; and

(ii) if the data required for an access lies within more than one valid abstract register, data is combined from those valid abstract registers to perform the access.

Claim 10 (original) A system for generating an intermediate representation of program code written for running on a programmable machine, the system comprising:

means for generating a plurality of register objects for holding variable values to be generated by the program code; and

means for generating a plurality of expression objects representing fixed values and/or relationships between said fixed values and said variable values according to said program code;

wherein at least one variably sized register is represented by plural register objects, one register object being provided for each possible size of the variably sized register.

Claim 11 (original) A system for generating an intermediate representation of program code expressed in terms of the instruction set of a subject processor comprising of at least one variably sized register, the system comprising:

means for generating a set of associated abstract register objects representing the variably sized register;

means for writing, for each write operation of a certain field width to the variable sized register to an abstract register object of the same width;

means for maintaining a record of which abstract register objects contain valid data, the record being updated upon each write operation; and

means for determining from said record, for each read operation of a given width, whether there is valid data in more than one of said different sized abstract registers of the set which must be combined to give the same effect as the same read operation performed upon the variable size register, and

(a) if it is determined that no combination is so required, reading directly from the appropriate register; or

(b) if it is determined that data from more than one register must be so combined, combining the contents of those registers.

Claim 12 (New) The method of claim 1, wherein said variably sized register is represented by separately addressable subsets of register objects.

Claim 13 (New) The method of claim 12, wherein said separately addressable subsets of register objects concurrently represent the same variably sized register.

Claim 14 (New) The method of claim 13, wherein partially redundant expressions are concurrently attached to more than one of said separately addressable subsets of register objects.

Claim 15(New) The method of claim 8, wherein said variably sized register is represented by separately addressable subsets of abstract register objects.

Claim 16 (New) The method of claim 15, wherein said separately addressable subsets of abstract register objects concurrently represent the same variably sized register.

Claim 17 (New) The method of claim 16, wherein partially redundant expressions are concurrently attached to more than one of said separately addressable subsets of abstract register objects.

Claim 18 (New) The method of claim 10, wherein said variably sized register is represented by separately addressable subsets of register objects.

Claim 19 (New) The method of claim 18, wherein said separately addressable subsets of register objects concurrently represent the same variably sized register.

Claim 20 (New) The method of claim 19, wherein partially redundant expressions are concurrently attached to more than one of said separately addressable subsets of register objects.